

Adaboost aplicado a clasificación de fonemas

Elkin Eduardo García Díaz

Abstract—Este artículo estudia la información contenida en fragmentos digitales de voz humana. Los fragmentos son procesados usando la transformada Wavelet con el fin de encontrar parámetros representativos en un fonema. Estos parámetros son entradas a un clasificador implementado con el algoritmo Adaboost usando como algoritmo débil un perceptrón .

This paper studies the information in digital frames of human voice. The frames are processed using wavelet transform with the purpose for finding representative parameters in a phoneme. These parameters are inputs on a classifier implemented with Adaboost algorithm using a perceptron as weak learner

Index Terms— Clasificación, Perceptrón, fonemas, Adaboost.

I. INTRODUCCIÓN

EL problema de reconocimiento de fonemas hace referencia a decidir cual es la identidad fonética de un sonido en una palabra. Es una de las alternativas para abordar el problema de reconocimiento de voz, y su aplicación está en todas aquellas áreas de investigación en torno a la interacción y comunicación oral hombre-máquina, siendo un campo importante de investigación por más de seis décadas [1].

Dentro de este campo de investigación, muchas técnicas han sido aplicadas para resolver eficientemente este problema. Robinson [2] creó un sistema para reconocer fonemas utilizando una red neuronal recurrente, una variante de la red neuronal estándar en donde las predicciones anteriores se retroalimentan a la red, incorporando mayor información.

Otros trabajos han hecho uso de *Support Vector Machines (SVM)*. Clarkson [3] creó un sistema multiclase para clasificación de fonemas, asumiendo conocidas las fronteras del fonema pero obteniendo un resultado inferior respecto a [2] a pesar de trabajar con un problema más sencillo. En [4] se desarrolló un sistema híbrido entre *SVM* y *Modelos ocultos de Markov (HMM)* por sus siglas en inglés) donde las predicciones de *SVM* eran entrada al *HMM*.

Es importante destacar que estos trabajos han tratado el problema de reconocimiento de fonemas pero en el idioma inglés, y aunque existen similitudes con el idioma español, el problema a resolver varía en gran medida.

Adicionalmente, dada la complejidad del problema a abordar, se hace necesario imponer ciertas restricciones que simplifiquen y reduzcan la complejidad del problema, reduciendo el número de fonemas a clasificar y delimitando la duración y ubicación de los fonemas dentro de la señal (de forma similar a [3]), dentro de un ambiente controlado.

A continuación, en la sección II se revisan los conceptos básicos de la *transformada Wavelet* y de *clasificación*, así como los fundamentos de los algoritmos de *Boosting*, la sección III presenta el proceso de extracción de características y preprocesamiento, el clasificador propuesto usando *Adaboost* y el

análisis de los experimentos realizados a partir del clasificador de fonemas implementado. Finalmente la sección IV presenta las conclusiones.

II. PRELIMINARES Y NOTACIÓN

A. Transformada Wavelet

La transformada Wavelet proporciona información que relaciona tiempo y frecuencia, es decir qué bandas de frecuencias están presentes en determinado intervalo de tiempo. Es importante anotar que es imposible establecer en qué tiempo está presente una componente de frecuencia, pues mientras se tiene mayor precisión en el tiempo se pierde precisión en frecuencia, esto se conoce como el principio de incertidumbre [5].

Adicionalmente, la transformada Wavelet es multiresolución; es decir que analiza la señal a diferentes frecuencias con diferentes resoluciones, en este caso una alta resolución en tiempo y pobre resolución en frecuencia a frecuencias altas y baja resolución en tiempo y alta resolución en frecuencia a frecuencias bajas. [6].

La *Transformada Continua Wavelet (CWT)* por sus siglas en inglés) está definida para una señal $x(t)$ como

$$\gamma(\tau, s) = \int_{-\infty}^{+\infty} x(t) \frac{1}{\sqrt{|s|}} \overline{\psi\left(\frac{t-\tau}{s}\right)} dt \quad (1)$$

Donde τ o traslación hace referencia al desplazamiento de la función madre o prototipo $\psi(t)$ mientras que s o escala se refiere al ancho de dicha función. Es importante destacar que aunque no exista una variable frecuencia dentro de la definición de la transformada Wavelet, la variable escala se relaciona con este parámetro, donde altas escalas hacen referencia a bajas frecuencias mientras bajas escalas hacen referencia a altas frecuencias.

La función original $x(t)$ puede ser reconstruida con la transformada inversa

$$x(t) = \frac{1}{C_\psi} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \gamma(\tau, s) \psi\left(\frac{t-\tau}{s}\right) d\tau \frac{ds}{|s|^2} \quad (2)$$

Donde $C_\psi = \int_{-\infty}^{+\infty} \frac{|\hat{\psi}(\zeta)|^2}{|\zeta|} d\zeta$ es llamada la constante de admisibilidad y $\hat{\psi}$ es la transformada de fourier de ψ . Para que la transformada inversa exista, se debe cumplir que $0 < C_\psi < +\infty$ lo cual implica que $\hat{\psi}(0) = 0$ [7].

B. El problema de clasificación

Sea \mathcal{X} un espacio de entrada, \mathcal{Y} un espacio de etiquetas y Δ una distribución sobre \mathcal{X} y dada una secuencia $\mathcal{S} = \{\langle \mathbf{x}_i, y_i \rangle\}_{i=1}^m$ de ejemplos etiquetados donde cada $\mathbf{x}_i \in \mathcal{X}$ es independiente e idénticamente distribuido de acuerdo a Δ y cada $y_i \in \mathcal{Y}$ es asignado de acuerdo a una regla posiblemente

estocástica. Para el caso del problema de clasificación binaria se restringe a $\mathcal{Y} = \{-1, +1\}$.

Una *regla de clasificación* llamada *hipótesis*, es una función $h : \mathcal{X} \mapsto \mathcal{Y}$ que asigna una etiqueta a cada elemento en el espacio de entrada. En el problema de clasificación binaria se tiene $h : \mathcal{X} \mapsto [-1, +1]$, donde el signo de $h(\mathbf{x})$ es interpretado como la predicción de la etiqueta a ser asignada a la instancia \mathbf{x} , mientras que la magnitud $|h(\mathbf{x})|$ es interpretada como la “confianza” de esta predicción. Por otra parte una clase de hipótesis \mathcal{H} es un conjunto compuesto por diferentes hipótesis en el espacio de entrada.

El desempeño de una hipótesis será evaluado utilizando el error de generalización R y el error empírico R_{emp} definidos como:

$$R(h) = \mathbb{P}_{(\mathbf{x}, y \sim \Delta)} \{ \text{sgn}(h(\mathbf{x})) \neq y \} \quad (3)$$

$$R_{emp}(h, \mathcal{S}, \mathcal{D}) = \sum_{i=1}^m D(i) \llbracket \text{sign}(h(\mathbf{x}_i)) \neq y_i \rrbracket \quad (4)$$

Donde $D \in \mathbb{R}^m$ es una distribución discreta sobre el conjunto de muestras etiquetadas y $\llbracket \cdot \rrbracket$ es la función indicadora.

Un *algoritmo de aprendizaje* es un procedimiento eficiente que toma como entradas un conjunto de muestras etiquetadas \mathcal{S} y una distribución discreta $D \in \mathbb{R}^m$ para retornar una hipótesis $h \in \mathcal{H}$. Un *clasificador combinado* $H(\mathbf{x})$ es la combinación convexa de varias hipótesis h_i de tal forma que

$$H(\mathbf{x}) = \sum_{i=1}^T \alpha_i h_i(\mathbf{x}) \quad (5)$$

Donde $\alpha_i \geq 0$ y $\sum_{i=1}^T \alpha_i = 1$. Cada hipótesis h_i se conoce como *clasificador base*.

C. Boosting

Las estrategias de Boosting pretenden elevar el desempeño de un algoritmo de aprendizaje débil combinando varias hipótesis adecuadamente generando un algoritmo de aprendizaje fuerte. El algoritmo Adaboost [8] o Boosting adaptativo introducido en [9] es un meta-algoritmo (un procedimiento que usa otro procedimiento como subrutina) que toma un conjunto de muestras etiquetadas \mathcal{S} , una distribución discreta D y un aprendiz débil *Weak* para retornar un clasificador combinado en T iteraciones.

En cada iteración t Adaboost ejecuta *Weak* sobre el conjunto \mathcal{S} con la distribución D_t para obtener la hipótesis h_t . De acuerdo al desempeño de h_t sobre el conjunto \mathcal{S} se obtiene un coeficiente α_t , que indica el peso de esa hipótesis dentro del clasificador combinado H . Así mismo el algoritmo modifica D_{t+1} para la siguiente iteración de acuerdo a

$$D_{t+1}(i) = \frac{D_t(i) \exp(-\alpha_t y_i h_t(x_i))}{Z_t} \quad (6)$$

Donde Z_t es un factor de normalización para que D_{t+1} sea una distribución. De un análisis de 6 se destaca que $y_i h_t(x_i)$ es positivo si el dato x_i está bien clasificado de acuerdo a h_t y negativo en caso contrario, de esta forma se le da menor peso

Algorithm: Adaboost ($\mathcal{S}, D_1, T, \text{Weak}$)

Input: $\mathcal{S} = \{x_i, y_i\}_{i=1}^m, D_1, T, \text{Weak}(\cdot, \cdot)$

Output: $H(\cdot)$

for $t = 1$ **to** T **do**

Obtener una hipótesis débil usando D_t .

$h_t \leftarrow \text{Weak}(\mathcal{S}, D_t)$.

Escoger $\alpha_t \in \mathbb{R}$. Usualmente:

$$\alpha_t = \frac{1}{2} \ln \left(\frac{1 - R_{emp}(h_t, \mathcal{S}, D_t)}{R_{emp}(h_t, \mathcal{S}, D_t)} \right). \quad (7)$$

Actualizar D_{t+1}

end

Retornar la hipótesis final:

$$H(x) = \text{sign} \left(\sum_{t=1}^T \frac{\alpha_t}{\sum_t \alpha_t} h_t(x) \right). \quad (8)$$

Fig. 1: Algoritmo Adaboost.

a las muestras bien clasificadas y mayor peso a las muestras mal clasificadas con el objetivo que el siguiente clasificador se concentre en estas últimas, maximizando la cantidad de información que obtendrá en la siguiente ronda. En la figura 1 se muestra este algoritmo como fue presentado en [8].

III. IMPLEMENTACIÓN Y DESEMPEÑO

En esta sección se describen las diferentes etapas de desarrollo del clasificador: Construcción de la base de datos, extracción de características y preprocesamiento, entrenamiento e implementación y evaluación.

A. Construcción de la Base de Datos

El clasificador implementado, reconoce los fonemas /a, /e y /i. Tanto para la fase de entrenamiento como para la fase de evaluación, se hace necesario la construcción de una base de datos, la cual consta de 51 archivos en formato .wav a 22050Hz, monofónicos y con duración entre 1,5 s. y 5 s. distribuidos en grupos de 17 para cada uno de los fonemas.

Sin embargo, dado que para la extracción de parámetros era necesario un tiempo menor del fonema, de cada archivo se extrajeron 3 muestras de fonema diferentes. Dada la importancia de la selección adecuada de éstas en la parte de entrenamiento y de evaluación del clasificador, se incorporó el conocimiento previo teniendo en cuenta que en un archivo solo se presenta un fonema (se supone el archivo fragmentado de forma adecuada o grabado con esta característica). Por esta razón la muestra se toma lo más cercano posible del centro con el objetivo de eliminar posibles silencios de la señal o ruidos debidos a problemas de grabación. Adicionalmente la muestra debe cumplir unos requisitos mínimos de valor RMS y varianza, para descartar señales con poco volumen y señales puramente DC.

Entonces la base de datos definitiva consta de 153 fragmentos de fonemas distribuidos uniformemente, cada uno de 1000 muestras (45.35ms). Este número de muestras es experimental dado que tras varias pruebas se determinó que una persona es

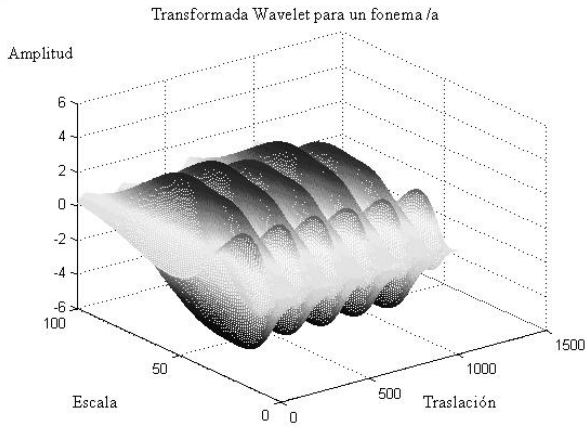


Fig. 2: Transformada Wavelet para un fonema /a

capaz de identificar dichos fonemas escuchando fragmentos de esta longitud.

En este punto se divide la base de datos en un conjunto de prueba y un conjunto de evaluación con 120 y 33 elementos respectivamente, donde las etiquetas están distribuidas uniformemente para cada fonema

B. Extracción de características y Preprocesamiento

Para la extracción de características, es de utilidad poseer información que relacione tiempo y frecuencia para una señal dada. En una señal estacionaria, el contenido de frecuencia es invariante en el tiempo, sin embargo este no es el caso de las señales de audio, como la música y la voz, razón por la cual es útil establecer una relación de este tipo.

Con este objetivo se descompone la señal original en varias componentes, para esto se hace uso de la *CWT* que utiliza como función madre al sombrero mejicano dado por

$$\psi(x) = \left(\frac{2}{\sqrt{3}} \pi^{-\frac{1}{4}} \right) (1 - x^2) e^{-\frac{x^2}{2}} \quad (9)$$

Esta función es proporcional a la segunda derivada de la función de densidad de probabilidad gaussiana [7]

En general, para los fonemas estudiados se presentan 2 grandes máximos en el eje escala, periódicos a lo largo del eje traslación, un ejemplo de ello se aprecia en la figura 2, donde se presenta la transformada Wavelet para un fonema /a. Por esta razón cada fonema se descompone solo en 2 patrones.

El algoritmo de *Descomposición de Señales en Patrones (DSAP)* usa como idea original la extracción de características detallada en [10] y utilizada en señales de señales musicales, este proceso se modifica como se muestra en la figura 3.

Este algoritmo está inspirado en el proceso de ortogonalización de *Gram-Schmidt* [11], pero la forma de hallar las bases es diferente, razón por la cual no necesariamente se obtienen señales ortogonales.

Pruebas empíricas permiten destacar que para las señales de los fonemas de la base de datos, los 2 primeros patrones x_{p1} y x_{p2} son suficientes para reconstruir con gran precisión la señal original, de la misma forma en que por un método similar se hallaban 2 patrones rítmicos en el caso de música

Algorithm: DSAP ($x(t), n, \psi(t)$)

Input: $x(t), n, \psi(t)$

Output: $x_{p1}(t), x_{p2}(t), \dots, x_{pn}(t)$

for $i = 1$ **to** n **do**

- Encontrar la *CWT* $\gamma(\tau, s)$ para $x(t)$ dado un $\psi(t)$ en un rango determinado de τ y s .

- Hallar la energía por escala $E(s) = \int_{\tau} \|\gamma(\tau, s)\|^2$.

- Definir s_{max} como el valor de escala donde la energía $E(s)$ es máxima.

- Hallar el patrón

$$x_{pi}(t) = \gamma(\tau, s_{max}) \quad (10)$$

- Redefinir $x(t)$

$$x(t) = x(t) - \langle x(t), x_{pi}(t) \rangle x_{pi}(t) \quad (11)$$

end

Retornar los patrones finales $x_{p1}(t), x_{p2}(t), \dots, x_{pn}(t)$

Fig. 3: Algoritmo de extracción de Patrones.

[10]. Esta comprobación no solo es auditiva al ser posible reconocer el fonema con la señal reconstruida, sino también comparando la señal punto a punto. Una característica interesante para destacar es que a pesar que el proceso no garantiza ortogonalidad entre los patrones obtenidos, se comprueba a partir de la realización del producto punto entre ellos que casi lo son.

Para la implementación del clasificador se extraen un total de 10 características enumeradas a continuación:

- Escalas de las cuales se extraen los elementos de la base. (2)
- Proyecciones de cada elemento de la base sobre la señal original. (2)
- Del análisis en frecuencia de cada patrón:
 - Varianza de la transformada de Fourier de cada elemento de la base. (2)
 - Ubicación del máximo pico de la transformada de Fourier de cada elemento de la base. (2)
- A partir de la correlación de los 2 patrones
 - Fase de la correlación de los elementos de la base. (1)
 - Máximo de la correlación de los elementos de la base. (1)

Estas características buscan extraer información tanto en tiempo como en frecuencia de cada uno de los patrones hallados, así como de la relación que existe entre los 2. Adicionalmente, es el resultado del trabajo previo detallado en [10] donde inicialmente se proponían 33 parámetros. Sin embargo algunos de ellos a pesar de ser útiles eran redundantes, lo que se podía comprobar al reducir la dimensionalidad por medio de la transformación de *Karhunen-Loeve (KLT)* [12].

Antes que los datos entren al clasificador, se realiza un pre-procesamiento de estas características con el objetivo de descartar contenido poco informativo, para ello se utiliza la transformación de *(KLT)* [12]. En el nuevo espacio de coordenadas que ha sido generado, los datos tienen una media de cero y una matriz de covarianza dada por la identidad. A partir de los valores propios de dicha matriz, se descartan los

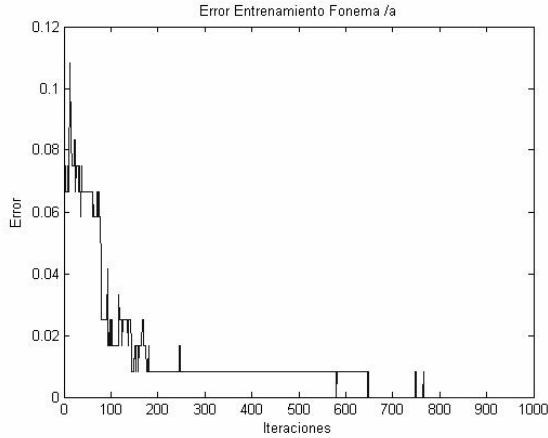


Fig. 4: Error de entrenamiento fonema /a

de valor más pequeño, pues su contribución de información no es relevante. De allí se reduce a 7 el número de entradas al clasificador.

C. Entrenamiento e Implementación del Clasificador

El clasificador débil utilizado para el entrenamiento es un perceptrón con 8 entradas (los 7 parámetros x_i y el bias b) y una salida $f(\mathbf{x})$, el limitador de dicho perceptrón es una sigmoide. La función de este clasificador está dada por

$$f(\mathbf{x}) = \text{tansig} \left(\sum_{i=1}^7 (\omega_i x_i) + b \right) \quad (12)$$

El objetivo del entrenamiento de este clasificador débil, es encontrar los valores adecuados de ω_i y b para que el error sea mínimo en el conjunto de entrenamiento.

Debido a que el problema de clasificación que se quiere resolver es un problema de múltiples clases pero de una sola etiqueta y que el clasificador escogido es binario, se tratan como 3 problemas monoetiqueta de una sola clase, donde las clases son /a, /e, e /i.

Cada uno de estos problemas se resuelve utilizando Adaboost realizando 1000 iteraciones de Adaboost para cada clase, los resultados del entrenamiento se presentan en las figuras 4, 5 y 6

Es claro que el concepto más difícil de aprender es el fonema /e, para el cual el error no llegó a cero, mientras que para los otros 2 fonemas sí. A pesar de alcanzar error 0 en los fonemas /a e /i se sigue entrenando, esto con el objetivo de aumentar la confianza de la clasificación, característica fundamental de *Adaboost* [8].

Para hallar la hipótesis final h_T a partir de las hipótesis H_1 , H_2 y H_3 obtenidas de la aplicación de boosting de acuerdo a la combinación de aprendices débiles h_{ij} dada por (8) se utiliza

$$h_T(x) = \text{máx}(f_1(x), f_2(x), f_3(x)) \quad (13)$$

Donde $f_j(x)$ es la salida del clasificador total H_j sin el limitador duro

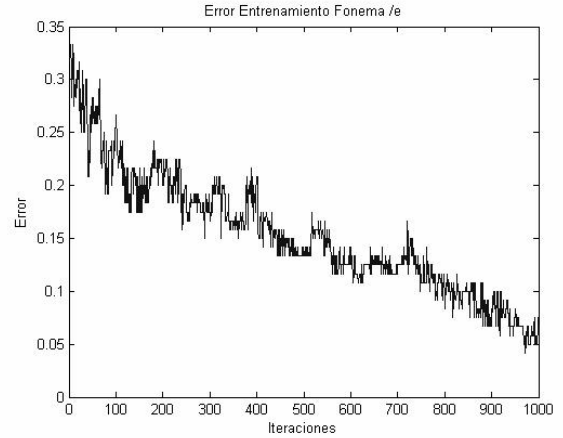


Fig. 5: Error de entrenamiento fonema /e

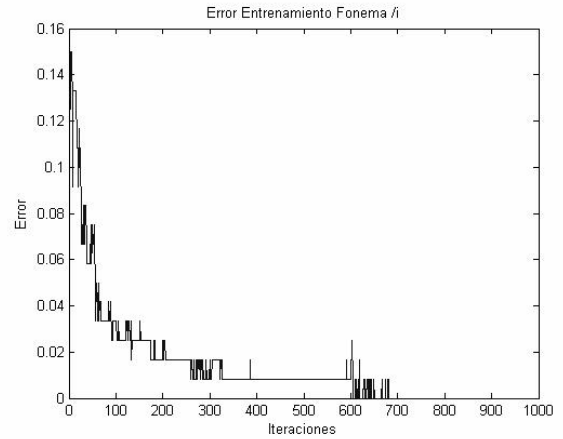


Fig. 6: Error de entrenamiento fonema /i

$$H_j(x) = \text{sgn}(f_j(x)) = \frac{\sum_i \alpha_{ij} h_{ij}(x)}{\sum_i \alpha_{ij}} \quad (14)$$

De acuerdo a (13) en caso que más de un clasificador dé etiqueta positiva, la etiqueta será la de mayor confianza; si por el contrario, ninguno tiene etiqueta positiva, la etiqueta será la de menor confianza. Teniendo esto en cuenta, el error de entrenamiento total del clasificador de fonemas también llega a 0 como se muestra en la figura 7

D. Evaluación del Clasificador Implementado

Usando el clasificador obtenido en la sección III-C se evalúa el desempeño de éste, por medio del conjunto de evaluación descrito en la sección III-A y que no ha sido usado en el proceso de entrenamiento. El error de evaluación del clasificador también disminuye a medida que aumentan el número de rondas, como se aprecia en la figura 8 y a pesar de no llegar a 0, presenta un buen desempeño.

La tabla comparativa I muestra los resultados de desempeño del clasificador obtenido con *Adaboost*, destacándose que tanto para cada clase por separado, como para el clasificador total implementado, los resultados obtenidos son bastante competitivos, teniendo en cuenta aspectos como el número

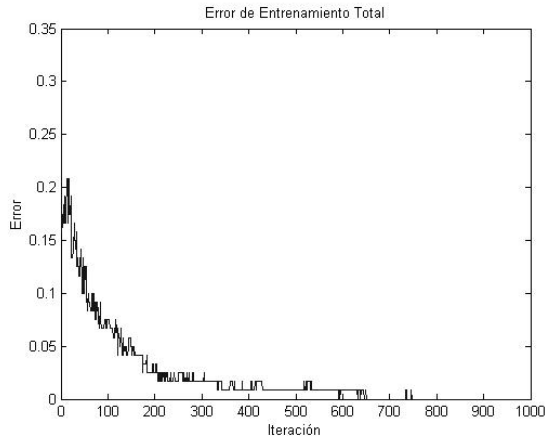


Fig. 7: Error de entrenamiento Total

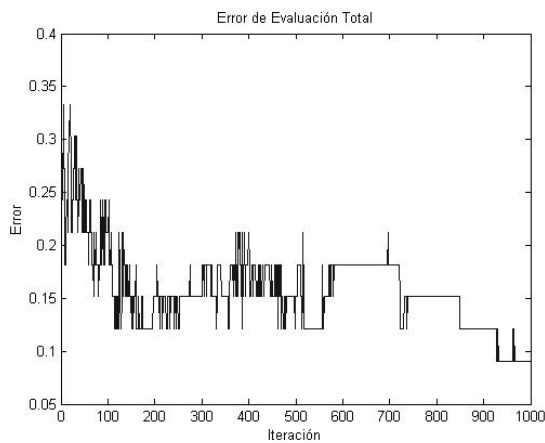


Fig. 8: Error de evaluación Total

de elementos del conjunto de entrenamiento, el cual es bajo y podría mejorarse para incluir muestras más representativas; la complejidad del clasificador, la corta longitud de la señal de entrada y la complejidad del problema, entre otros. A pesar que no son comparables los resultados obtenidos con el desempeño de [2], [3] y [4] no solo por el número de clases que allí se trabajaron y las características de sus bases de datos, sí se puede destacar que porcentualmente los resultados fueron mejores.

IV. CONCLUSIONES

La implementación de este clasificador de fonemas, permite corroborar la versatilidad y buen desempeño de los algoritmos de *Boosting*, facilitando la obtención de clasificadores sencillos, con un bajo costo computacional, ideal a la hora

TABLE I: Tabla Comparativa de Resultados de Entrenamiento y Evaluación

Clasificación	Fonema /a	Fonema /e	Fonema /i	Total
Correcta				
Entrenamiento	100 %	94.17 %	100 %	100 %
Evaluación	90.91 %	90.91 %	93.94 %	90.91 %

de implementarse directamente en hardware. Así mismo, la descomposición del problema multiclase a varios problemas de una clase permitió el tratamiento de este problema de forma satisfactoria, escogiendo de forma adecuada la hipótesis final a partir de los 3 clasificadores iniciales. Adicionalmente, AdaBoost permitió la construcción de un clasificador robusto para este problema, logrando la generalización deseada sin sobre ajuste a los datos, favoreciendo el uso de esta clase de algoritmos en aplicaciones donde existen elementos de la base de datos que presentan ruido, caso particular de la mayoría de señales de audio.

Por otra parte, el éxito del clasificador implementado se debe a la adecuada extracción de características a partir de la transformada Wavelet y al preprocesamiento, destacándose la forma novedosa y efectiva de descomponer la señal en varios patrones, lo que permite no solo analizar cada patrón por separado, sino también estudiar las relaciones que existen entre ellos, siendo más útil la información obtenida.

Quedan planteados como trabajos futuros, el verificar los resultados de estas pruebas en un conjunto de datos más grande y con más clases, para ver hasta qué punto se puede mantener la simplicidad del clasificador, así como continuar en el estudio teórico y práctico de la descomposición en patrones propuesta a partir de la transformada Wavelet. Adicionalmente se espera implementar esta clase de clasificadores, directamente en Hardware de aplicación específica

REFERENCES

- [1] R. Cole, J. Mariani, H. Uszkoreit, A. Zaenen, and V. Zue, "Survey of the state of the art in human language technology," 1995. [Online]. Available: citeseer.ist.psu.edu/article/cole97survey.html
- [2] A. J. Robinson, "An application of recurrent nets to phone probability estimation," *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 298–305, March 1994. [Online]. Available: citeseer.ist.psu.edu/robinson94application.html
- [3] P. Clarkson and P. Moreno, "On the use of support vector machines for phonetic classification," in *ICASSP'99*, vol. II, 1999.
- [4] A. Ganapathiraju, J. Hamaker, and J. Picone, "Hybrid svm/hmm architectures for speech recognition," 2000. [Online]. Available: citeseer.ist.psu.edu/article/picone00hybrid.html
- [5] C. S. Burrus, R. A. Gopinath, and H. Guo, *Introduction to Wavelets and Wavelet Transforms: a Primer*. Upper Saddle River, New Jersey: Prentice Hall, 1998.
- [6] C. K. Chui, *Wavelets : a mathematical tool for signal processing*. Philadelphia, PA: Society for Industrial and Applied Mathematics (SIAM), 1997.
- [7] I. Daubechies, *Ten Lectures on Wavelets*. Philadelphia, PA: Society for Industrial and Applied Mathematics (SIAM), 1992.
- [8] R. E. Schapire and Y. Singer, "Improved boosting algorithms using confidence-rated predictions," *Machine Learning*, vol. 37, no. 3, pp. 297–336, Dec. 1999. [Online]. Available: <http://www.boosting.org/papers/SchSin99b.ps.gz>
- [9] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 119–139, Aug. 1997. [Online]. Available: <http://www.boosting.org/papers/FreSch97.ps.gz>
- [10] E. García, J. Pacheco, and A. Mancera, "Clasificación de música por género utilizando redes neuronales artificiales," in *CIC*, O. Velez-Langs and C. A. Peña-Reyes, Eds. Universidad del Sinu, 2005, pp. 284–291.
- [11] G. Strang, *Introduction to Applied Mathematics*. Wellesley, MA: Wellesley-Cambridge Press, 1986.
- [12] K. Fukunaga, *Introduction to statistical pattern recognition (2nd ed.)*. San Diego, CA, USA: Academic Press Professional, Inc., 1990.